Lab 2

Title: Introduction to DCL AND TCL statements in SQL

Objective:

- ✓ To be familiar with DCL and TCL statements in SQL

Theory:

DCL (Data Control Languages)

- ✓ DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.
- ✓ It used to give and withdraw specific privileges (as defined by query) to the user in a multi-user database.
- ✓ By setting up the permission, user can prevent unauthorized access to the database.

DCL commands are:

- GRANT
- REVOKE

Before discussing GRANT and REVOKE let us consider the following relation.

employee_info(eid,name,address,department,salary)

GRANT:

- ✓ This is a SQL command which is used to provide privileges/permissions to modify and retrieve database objects like tables, views, indexes etc.
- ✓ It can be used to grant SELECT, INSERT, UPDATE, DELETE etc. privileges to a user.

Syntax:

GRANT <privilege list> on <relation or view> to <user>;

Example:

- ✓ GRANT INSERT, SELECT,UPDATE on employee_info to ram;
- ✓ GRANT ALL PRIVILEGES on employee_info to ram;
- ✓ GRANT INSERT on employee_info to hari;
- ✓ GRANT INSERT (eid,name,address) on employee_info to amit, ritu;
- ✓ GRANT UPDATE (eid,name,department) on employee_info to amit, ritu;
- ✓ GRANT DELETE on employee_info to hari;

REVOKE:

- ✓ It revokes the given access to the user.

syntax:

REVOKE<privilege list> on <relation or view> from <user>;

Example:

- ✓ REVOKE UPDATE on employee_info from ram;
- ✓ REVOKE SELECT on employee_info from ram;
- ✓ REVOKE INSERT (eid,name,address) on employee_info from amit, ritu;
- ✓ REVOKE UPDATE (eid,name,department) on employee_info from amit, ritu;
- ✓ REVOKE DELETE on employee_info from hari;

TCL (Transaction Control Language)

- ✓ Transaction Control Language (TCL) is a set of special commands that deal with the transactions within the database.
- ✓ Basically, they are used to manage transactions within the database.
- ✓ TCL commands are also used for maintaining the consistency of the database.
- ✓ These commands are generally used along with the DML commands such as INSERT, UPDATE and DELETE.
- ✓ The changes made by DML commands are either committed or rolled back by TCL commands.
- ✓ There is another TCL command that can place a save point in the transactions which makes it possible to rollback all the transaction till the last save point.

COMMIT:

Commit command make the changes made to the database permanent.

Syntax:

```
COMMIT;
```

Here's the syntax demonstrating the use of the COMMIT command with a transaction in MySQL:

```
START TRANSACTION;
{a set of SQL statements};
COMMIT;
```

The parameters used in the syntax are:

- ✓ START TRANSACTION: It is used for marking the beginning of changes or operations in a transaction.
- ✓ {a set of SQL statements}: It is used for mentioning the task that is supposed to be completed.
- ✓ COMMIT: It is used to save transactional changes made by SQL statements.

Example:

```
START TRANSACTION;

DELETE FROM student_info

WHERE sid = 11;

COMMIT ;
```

ROLLBACK:

- ✓ Rollback command is used to undo the changes that have been made to the database temporarily.
- ✓ The important point to note here is that the changes saved using COMMIT command cannot be undone using ROLLBACK command.

Example:

```
UPDATE student_info SET location='Dharan' WHERE name='ram';

ROLLBACK;
```

SAVEPOINT:

It's used to roll back a transaction to a specific point rather than the complete transaction.

Syntax:

SAVEPOINT SavepointName;

- ✓ Among all transactions, this command is exclusively used to create SAVEPOINT.
- ✓ ROLLBACK is a command that is used to undo a set of transactions.

The syntax for rollback to savepoint command:

ROLLBACK TO SavepointName;

Example:

```
UPDATE student_info
SET program = 'BBA'
WHERE sid = 5;
Savepoint A;

UPDATE student_info
SET name = 'ram'
WHERE location = 'pokhara;
SAVEPOINT B;
```

Now if we want to roll back the certain DML commands, we can do so by using Rollback like this:
This will rollback the transaction till save point A:

```
Rollback to A;
```

Demonstration

- Open MySQL and Apache xampp control panel
- Open the command prompt
- Change directory to xampp\mysql\bin
- Now, xampp\mysql\bin>mysql –u root –p –h localhost and it will ask for password
- Press Enter
- Now, you can perform below operations

1) Create a database named eemc _db

```
create database eemc_db;
```

//Database named eemc_db will be created

2) select database named eemc_db

```
use  eemc_db;
```

3) Create a table named employee_info with following columns and data type

Eid	Name	Address	department	salary
Int	varchar(30)	varchar(30)	varchar(30)	decimal (10,2)

```
create table employee_info(eid int,name varchar(30),address varchar(30),department
varchar(30),salary decimal (10,2));
```

For use of commit and rollback

3) Now insert some records into table named employee_info

Start transaction;

```
insert into employee_info values(1,'anish', 'kathmandu','civil',25000.35);

insert into employee_info values(2,'Roshan', 'pokhara','computer',18750.25);

insert into employee_info values(3,'rojina','kathmandu','computer',22250.45);

insert into employee_info values (4,'ramesh','bhaktapur','it',55250.15);
```

//you can see that no any changes is reflected in database while opening localhost/ phpmyadmin
//But changes is made locally, you can see this by using following query

```
select * from employee_info;
```

Now commit the transaction

commit ;

Now, you can see changes is reflected in database while opening localhost phpmyadmin

4)Now insert another 1 record

start transaction;

```
insert into employee_info values(5,'hari','pokhara','it',60250.65);
```

To see records

*select * from employee_info;*

Note: insertion is not reflected in database, which we can see through localhost/phpmyadmin

5) Now revert the operation of step(4)

rollback;

//Rollback operation will cancelled the above operation

To see records

select * from employee_info;

//We can see the previous record that is already committed

(similarly test with update and delete operations as well)

For use of savepoint

6)Update the address of employee to kathmandu whose name is 'anish'

Start transaction;

update employee_info set address='chitwan' where name='anish';

savepoint **update_anish;**

//We can see the information is updated but it is not reflected in database

7) **Delete the information of employee whose department is civil**

```
delete from employee_info where  department='civil';

savepoint delete_civil;
```

select *from employee_info;

//We can see the information is deleted but it is not reflected in database

8) **Rollback the transaction to step(8)**

```
Rollback to update_anish;
```

9) Commit the transaction

```
Commit;
```

//We can see that information of anish is updated and reflected in database but above deletion operation have been cancelled.

10) Create two users named Anish and Rita with following privilege to performing operations on database

Anish: SELECT, UPDATE, INSERT

Sita: SELECT, DELETE

Step1: Exit the maridb shell

exit;

Step2: Reconnect to the mariadb server

mysql -u root -p

Step3: create two users with name anish and sita

Syntax:

CREATE USER 'username' IDENTIFIED BY 'password';

```
CREATE USER 'anish'' IDENTIFIED BY 'anish123';
```

```
CREATE USER 'sita' IDENTIFIED BY 'sita123';
```

Step4: Give above mentioned privileges to two users with name anish and sita

GRANT SELECT,UPDATE,INSERT ON eemc_db.employee_info TO anish;

GRANT SELECT,DELETE ON eemc_db.employee_info TO sita;

Step5: Exit the mariadb shell

exit;

Step5: Reconnect mariadb server with user named anish

mysql -u anish –p

anish123

Step6: select the database named eemc_db

use eemc_db;

10) **Now, try to perform the above operations that is given privilege to user anish**

insert into employee_info values(6,'pradip','palpa','computer',4525.55);

(Insert operation will be performed successfully)

11) Try to perform the above operations that is not given privilege to user anish

delete from employee_info where eid=1;

(delete operation is not allowed for anish so it cannot be performed)

Now, switch to another user named sita

Step1: Exit the mariadb shell

exit;

Step2: Reconnect mariadb server with user named sita

mysql -u sita –p

sita123

Step3: select the database named eemc_db

use eemc_db;

12) **Now, try to perform the above operations that is given privilege to user sita**

delete from employee_info where eid=1;

(Delete operation is executed successfully for user sita)

13) **Try to perform the above operations that is not given privilege to user sita**

update employee_info set deparment='civil' where address='palpa';

(Update operation cannot be performed for user sita)

14) Revoke delete operation that is given to sita.

(Note you must switch with user root)

exit;

mysql -u root –p

revoke delete on eemc_db.employee_info from sita;

15)Now again try to perform delete operation by sita

(Note you must switch with user sita)

exit;

mysql –u sita -p

sita123;

use eemc_db;

delete from employee_info where eid=2;

(Now, delete operation cannot be performed for user sita)

Some useful commands:

To list out all users

select user from mysql.user;

To find out which grant is given for particular user

Syntax:

show grants for 'user_name' ;

Example:

show grants for 'root' ;

show grants for 'anish';

Discussion: (This portion is left for student)

Conclusion: (This portion is left for student)

*******THE END*******